|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Piotr Poleszak | |  |  | PSI | | | | GC02 | |
|  | AUTOR | ROK AKAD. |  | | | SPEC. |  | GRUPA |
| Analiza zbiorów danych z zastosowaniem algorytmów klasteryzacji | | | | |  | | | 2 | |
| TEMAT | | | NR SPRAWOZDANIA | |
| 05.11.2024 | LMAD | | | | |  | | | |
| DATA WYKONANIA | PRZEDMIOT | | | | | PROWADZĄCY | | | |

# **Cel**

Celem projektu jest przeprowadzenie analizy trzech różnych zbiorów danych z wykorzystaniem algorytmów klasteryzacji i technik przetwarzania danych. Analiza obejmuje:

* **Zadanie 1**: Segmentacja obrazu za pomocą algorytmu k-średnich.
* **Zadanie 2**: Klasteryzacja zbioru danych iris.csv przy użyciu algorytmów hierarchicznych i k-średnich.
* **Zadanie 3**: Grupowanie wydatków klientów hurtowych w Portugalii za pomocą algorytmu DBSCAN.

# **Materiały i metody**

### **2.1. Zbiory danych**

* **Zadanie 1**: Zdjęcie palm\_tree.jpg pobrane z Pexels.
* **Zadanie 2**: Zbiór danych iris.csv zawierający pomiary trzech gatunków irysów.
* **Zadanie 3**: Dane klientów hurtowych z UCI MLR

### **2.2. Metody**

#### Wczytywanie i przygotowanie danych

* **Biblioteki Python**: Użyto bibliotek **pandas**, **numpy**, **matplotlib**, **seaborn**, **scikit-learn** oraz **cv2** do manipulacji danymi, wizualizacji i implementacji algorytmów klasteryzacji.
* **Przetwarzanie danych**: Sprawdzono kompletność danych, wykonano standaryzację i normalizację tam gdzie było to wymagane.

#### Algorytmy klasteryzacji

* **Algorytm k-średnich**: Zastosowany do kwantyzacji kolorów w obrazie oraz klasteryzacji zbioru danych iris.csv.
* **Algorytmy hierarchiczne**: Użyto metod aglomeracyjnych z różnymi metodami łączenia (Warda, pojedynczego połączenia, całkowitego połączenia).
* **Algorytm DBSCAN**: Wykorzystany do identyfikacji klastrów i punktów odstających w danych klientów hurtowych.

#### Wizualizacja

* **Matplotlib i Seaborn**: Użyto do tworzenia wykresów i wizualizacji w celu interpretacji wyników klasteryzacji.

# **Wyniki i dyskusja**

### **3.1. Zadanie 1: Segmentacja obrazu za pomocą algorytmu k-średnich**

#### **3.1.1. Wyświetlenie kodów RGB dla każdego piksela**

Zaimportowano obraz palm\_tree.jpg i przekonwertowano go do formatu RGB. Kody RGB dla każdego piksela zostały wyświetlone w postaci macierzy.

#### **3.1.2. Wyświetlenie oryginalnego zdjęcia**

Oryginalne zdjęcie zostało wyświetlone przy użyciu biblioteki Matplotlib.



#### **3.1.3. Kwantyzacja kolorów za pomocą algorytmu k-średnich**

Przeprowadzono kwantyzację kolorów, stosując algorytm k-średnich z 6 klastrami. Dane obrazu zostały przekształcone z formatu 3D (szerokość, wysokość, kanały kolorów) do 2D (liczba pikseli, kanały kolorów), aby algorytm mógł je przetworzyć.

#### **3.1.4. Wyświetlenie rozmiarów zdjęcia i liczby kanałów kolorów (RGB)**

Rozmiary zdjęcia zostały określone na podstawie jego wymiarów: szerokość, wysokość oraz liczba kanałów kolorów.

#### **3.1.5. Przekształcenie danych z formatu 3D do 2D**

Dane obrazu zostały przekształcone do formatu 2D o wymiarach odpowiadających liczbie pikseli i liczbie kanałów kolorów.

#### **3.1.6. Zastosowanie algorytmu k-średnich dla 6 klastrów**

Algorytm k-średnich został zastosowany z parametrem n\_clusters=6. Pozwoliło to na podział kolorów w obrazie na 6 głównych grup.

#### **3.1.7. Wyświetlenie etykiet klastrów dla każdego piksela**

Etykiety klastrów zostały przypisane do każdego piksela w obrazie, tworząc macierz etykiet o rozmiarach odpowiadających wymiarom obrazu.

#### **3.1.8. Wyświetlenie współrzędnych centroidów**

Współrzędne centroidów zostały wyświetlone. Centroidy reprezentują dominujące kolory w obrazie po kwantyzacji.

#### **3.1.9. Zaokrąglenie współrzędnych centroidów**

Centroidy zostały zaokrąglone do liczb całkowitych, aby mogły reprezentować wartości RGB (zakres od 0 do 255).

#### **3.1.10. Utworzenie skwantyzowanego obrazu**

Każdy piksel w oryginalnym obrazie został zastąpiony odpowiadającym mu centroidem (średnim kolorem klastra), tworząc skwantyzowany obraz z ograniczoną paletą kolorów.

#### **3.1.11. Wyświetlenie obrazu po kwantyzacji**

Obraz po kwantyzacji został wyświetlony, ukazując efekt redukcji liczby kolorów do 6. Obraz zachowuje ogólny wygląd oryginału, ale z wyraźnie mniejszą liczbą odcieni.



**Dyskusja:**

Kwantyzacja kolorów zredukowała liczbę kolorów w obrazie, co pozwoliło na segmentację obrazu na podstawie podobieństwa kolorów. Jest to przydatne w kompresji obrazu oraz w uproszczeniu analizy obrazów np. w identyfikacji obiektów na podstawie koloru.

### **3.2. Zadanie 2: Analiza zbioru danych iris.csv**

#### **3.2.1. Wczytanie danych i sprawdzenie kompletności**

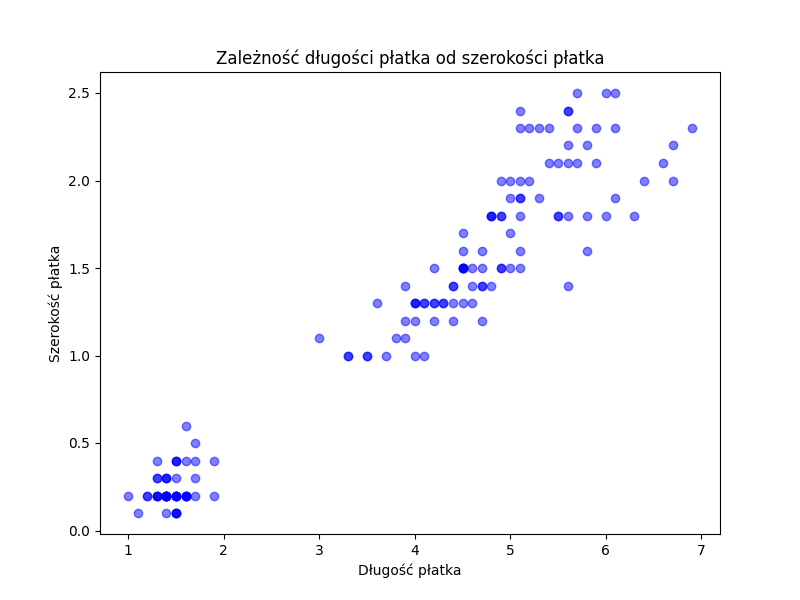
Dane zostały wczytane z pliku iris.csv. Sprawdzono, że nie zawierają brakujących wartości. Zbiór danych zawiera 150 obserwacji, każda z czterema cechami (długość i szerokość kielicha oraz płatka) oraz etykietą gatunku.

#### **3.2.2. Statystyki podstawowe i informacje o danych**

Przeanalizowano statystyki opisowe, takie jak średnia, odchylenie standardowe, wartości minimalne i maksymalne dla każdej cechy. Informacje o danych potwierdziły, że wszystkie kolumny są typu numerycznego, a kolumna 'species' jest typu kategorycznego.

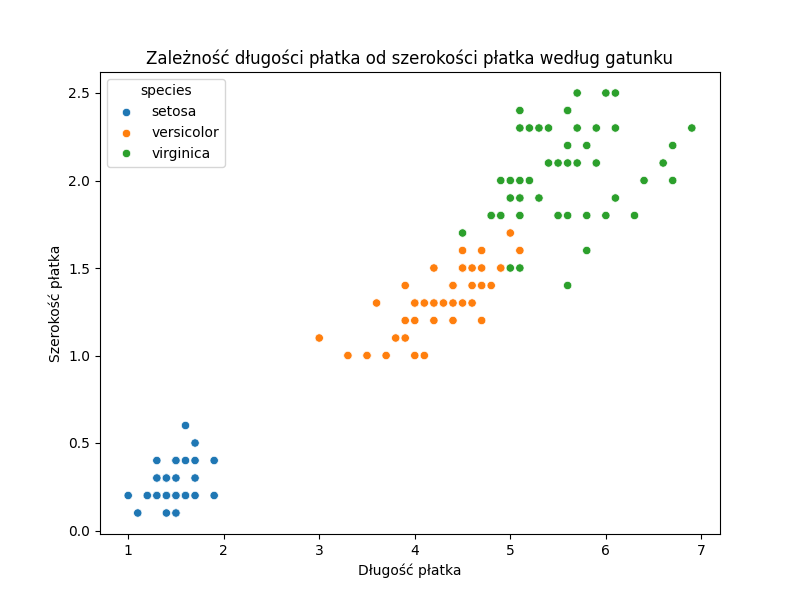
#### **3.2.3. Wykres punktowy zależności długości płatka w funkcji szerokości płatka**

Wykres punktowy pokazał zależność między długością a szerokością płatka dla wszystkich obserwacji. Dane naturalnie grupują się w trzy skupienia, sugerując możliwość podziału na trzy klastry.



#### **3.2.4. Wykres punktowy różnicowany za pomocą gatunku**

Dane zostały zwizualizowane z podziałem na gatunki, co uwidoczniło wyraźne rozdzielenie między trzema gatunkami irysów na podstawie cech płatka.

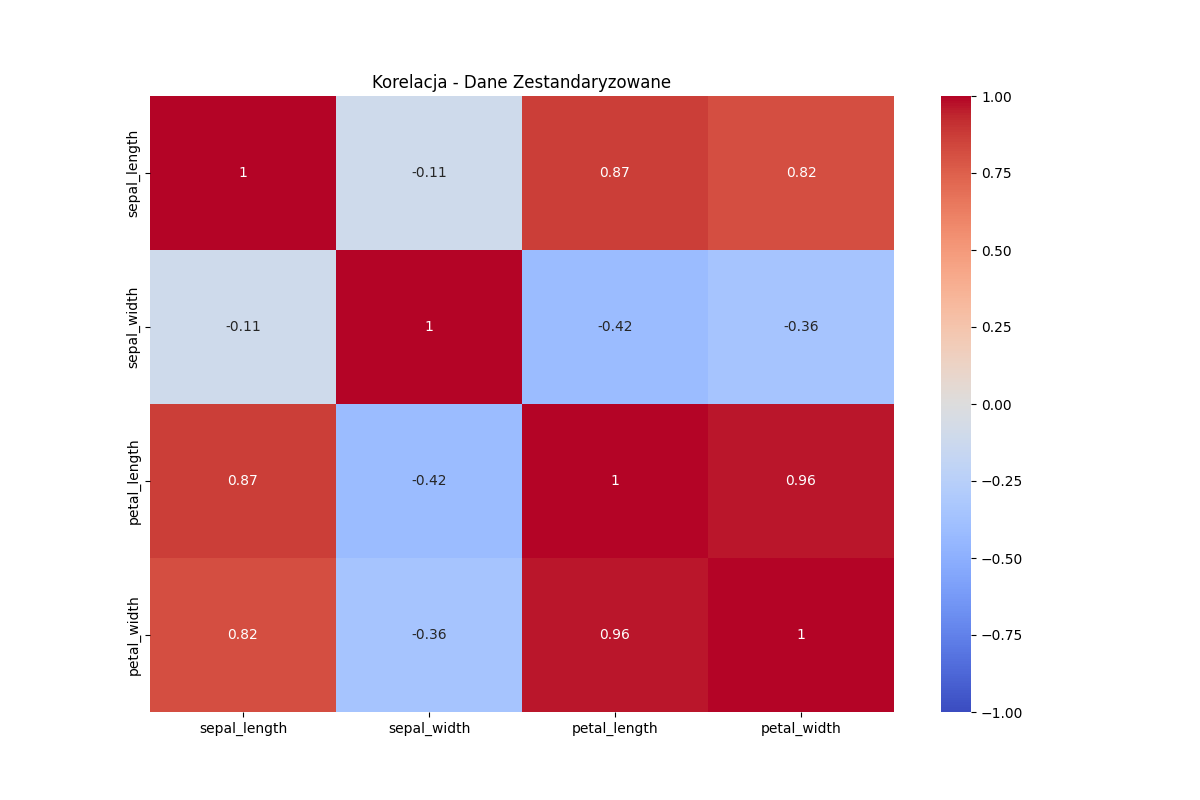
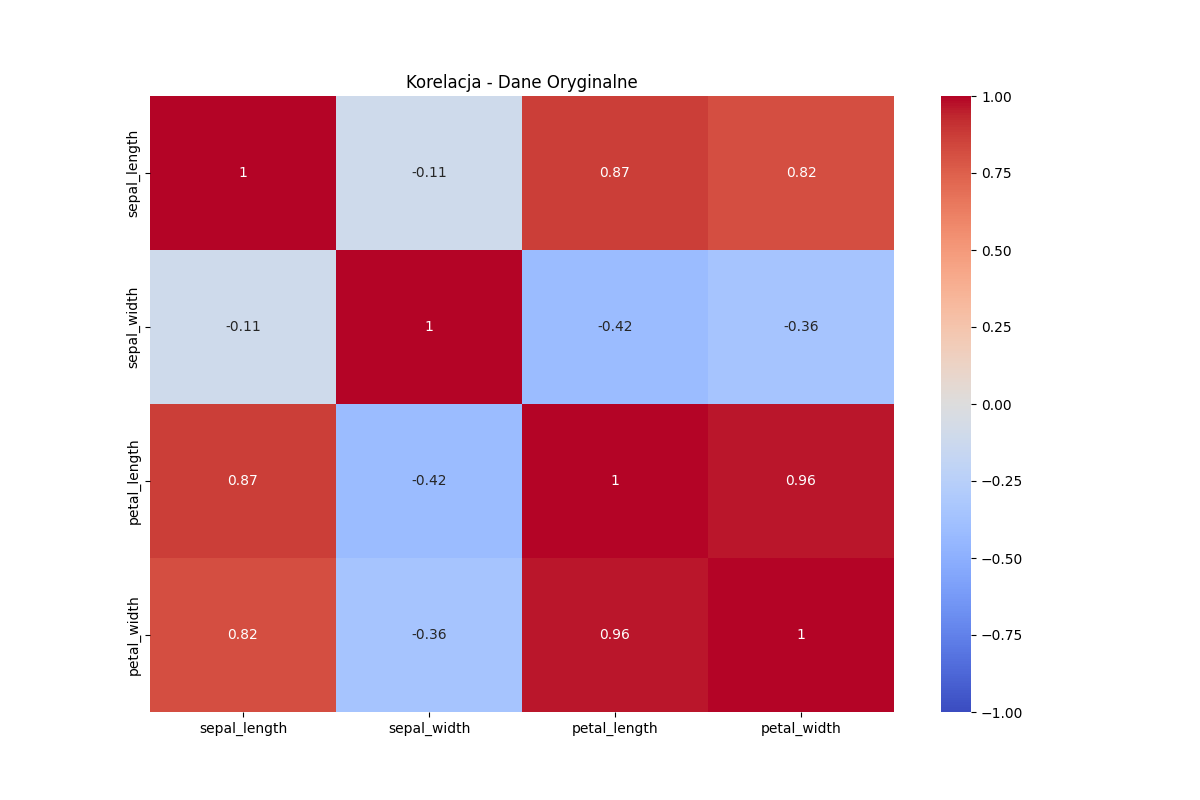


#### **3.2.5. Przygotowanie danych do klasteryzacji**

* **Usunięcie kolumny 'species'**: Aby skupić się na danych numerycznych.
* **Standaryzacja danych**: Dane zostały zestandaryzowane, aby każda cecha miała średnią zero i odchylenie standardowe jeden.
* **Normalizacja danych**: Dane zostały znormalizowane do zakresu [0,1].

#### **3.2.6. Heatmapy korelacji**

Utworzono mapy korelacji dla danych oryginalnych, zestandaryzowanych i znormalizowanych. Mapy te pokazały, że istnieje silna dodatnia korelacja między długością i szerokością płatka, a słabsza korelacja między cechami kielicha.



#### 

#### **3.2.7. Klasteryzacja danych oryginalnych**

**Algorytm hierarchiczny (metoda aglomeracyjna):**

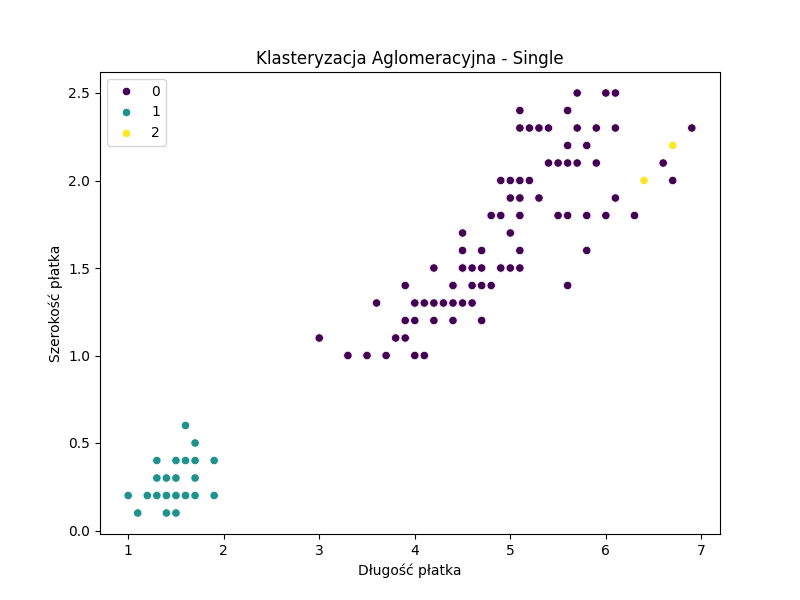
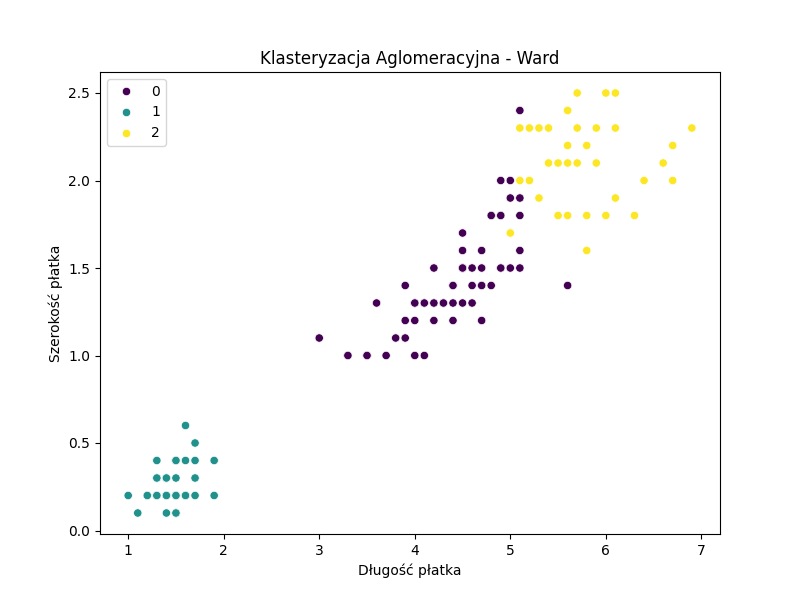
* Zastosowano trzy metody łączenia:
  + **Warda**
  + **Pojedynczego połączenia**
  + **Całkowitego połączenia**
* Dla każdej metody przeprowadzono klasteryzację na trzy klastry.
* Obliczono współczynnik sylwetki dla oceny jakości klasteryzacji.

**Algorytm k-średnich:**

* Zastosowano z parametrem n\_clusters=3.
* Obliczono współczynnik sylwetki.

#### **3.2.8. Wykresy punktowe z etykietami klastrów**

Dla każdej metody klasteryzacji utworzono wykresy punktowe, gdzie punkty były różnicowane za pomocą etykiet nadanych przez algorytmy.



#### 

#### **3.2.9. Porównanie wyników klasteryzacji z rzeczywistymi gatunkami**

* **Metoda Warda** i **algorytm k-średnich** dały wyniki klasteryzacji zbliżone do rzeczywistych gatunków.
* **Metody pojedynczego i całkowitego połączenia** miały gorsze wyniki, co potwierdziły niższe wartości współczynnika sylwetki.

#### **3.2.10. Klasteryzacja na danych zestandaryzowanych i znormalizowanych**

Proces klasteryzacji został powtórzony dla danych zestandaryzowanych i znormalizowanych. Standaryzacja danych poprawiła wyniki klasteryzacji, co objawiło się wyższymi wartościami współczynnika sylwetki.

**Dyskusja:**

Standaryzacja i normalizacja danych mają wpływ na wyniki klasteryzacji. W przypadku zbioru Iris standaryzacja pozwoliła na lepsze dopasowanie modeli klasteryzacji do naturalnej struktury danych.

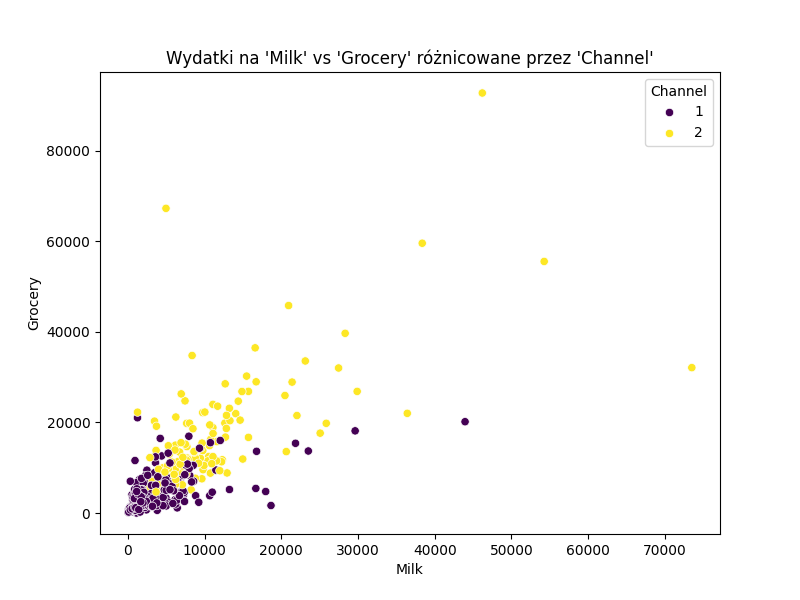
### **3.3. Zadanie 3: Grupowanie wydatków klientów hurtowych za pomocą algorytmu DBSCAN**

#### **3.3.1. Wczytanie danych i przygotowanie**

Dane zostały wczytane z pliku Wholesale Customers. Zawierają informacje o wydatkach klientów hurtowych na różne kategorie produktów oraz informacje o kanale sprzedaży i regionie.

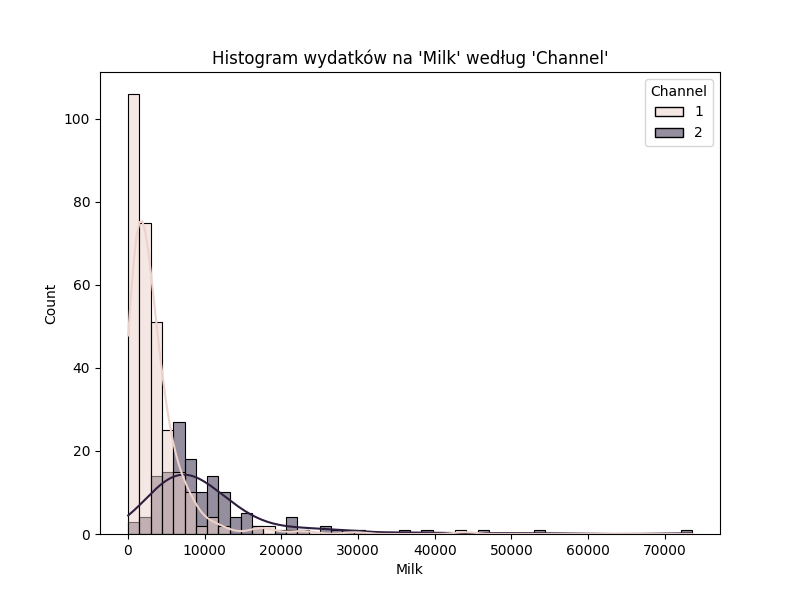
#### **3.3.2. Wykres punktowy 'Milk' vs 'Grocery' różnicowany przez 'Channel'**

Wykres pokazał zależność między wydatkami na mleko a wydatkami na artykuły spożywcze, różnicując punkty na podstawie kanału sprzedaży (hotel/ restauracja/ kawiarnia vs detaliści).



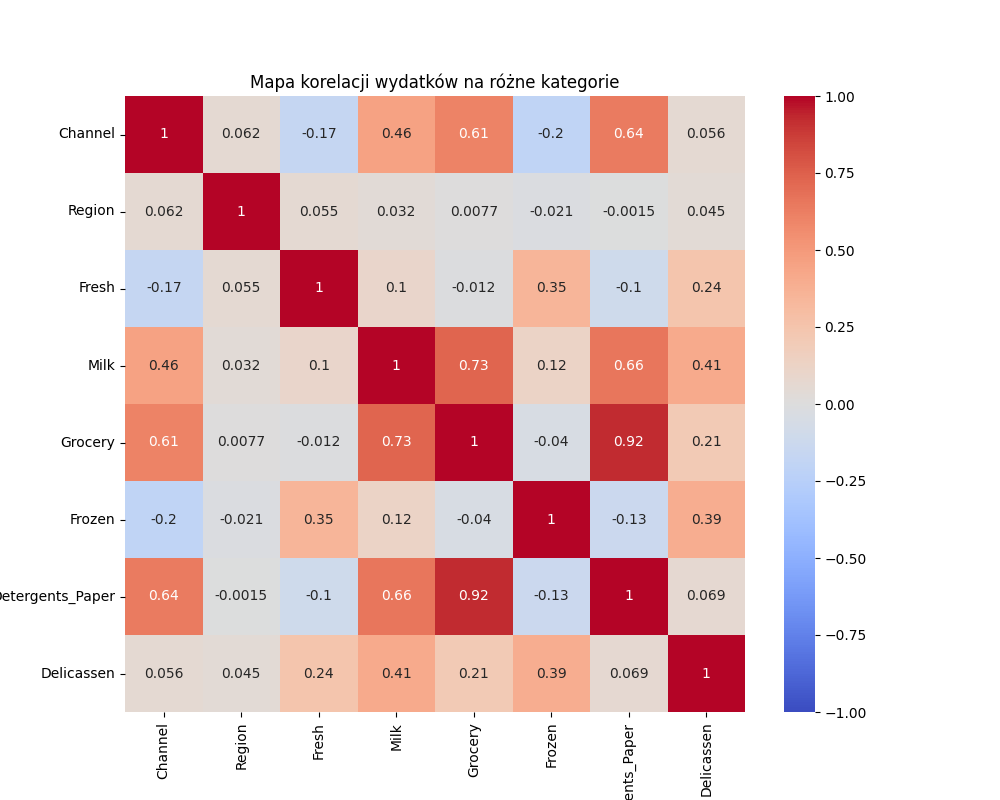
#### **3.3.3. Histogram wydatków na mleko na podstawie 'Channel'**

Histogram ukazał rozkład wydatków na mleko dla obu kanałów sprzedaży, pozwalając zauważyć różnice w zachowaniach zakupowych.



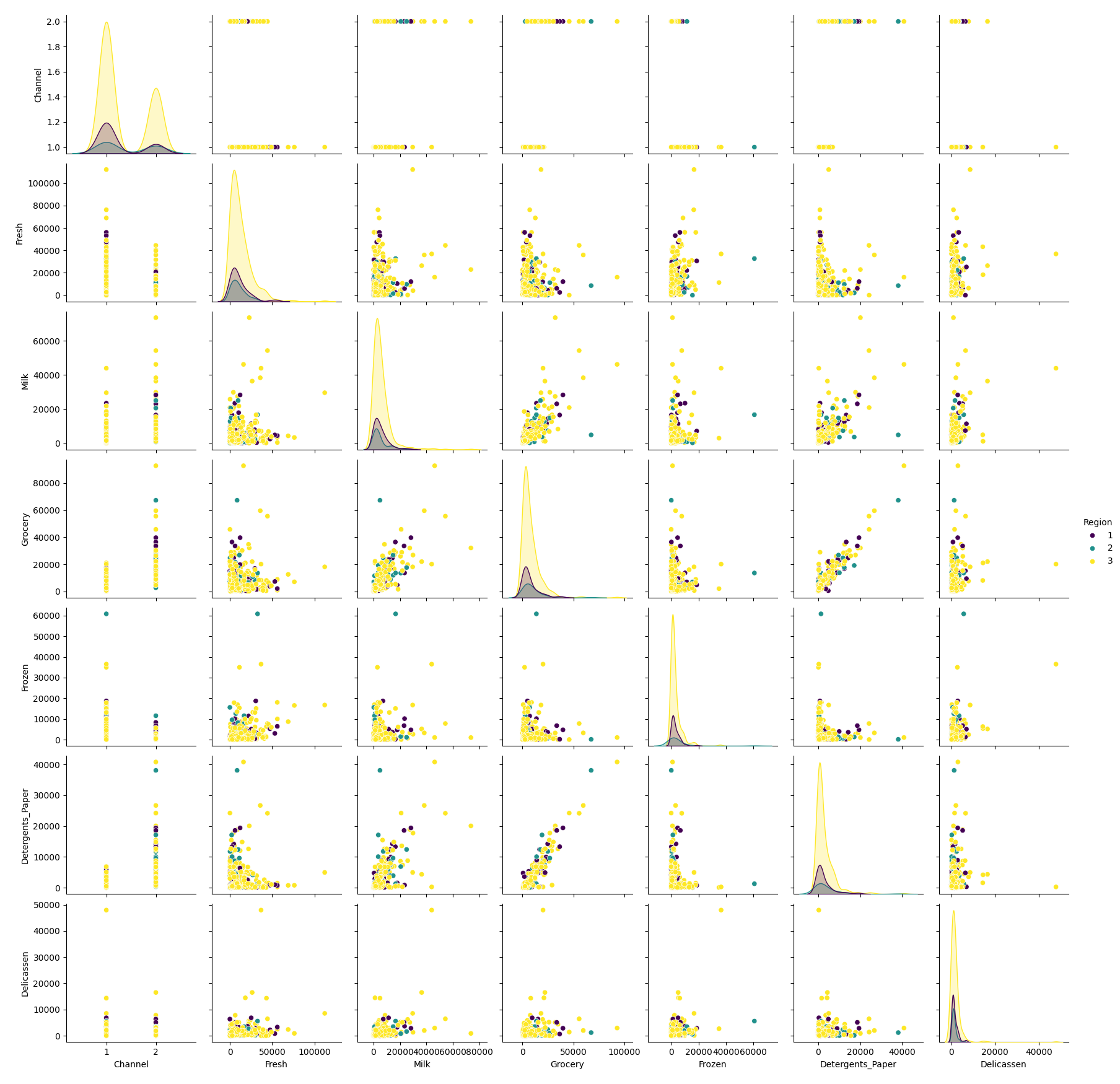
#### **3.3.4. Mapa korelacji wydatków**

Mapa korelacji między wydatkami na różne kategorie produktów wykazała, które kategorie są ze sobą powiązane. Na przykład, silna korelacja między wydatkami na artykuły spożywcze a detergentami.



#### **3.3.5. Macierz par zmiennych różnicowanych przez 'Region'**

Utworzono macierz par, aby zbadać zależności między wszystkimi parami zmiennych, różnicując punkty na podstawie regionu (Lizbona, Porto, inny).



#### **3.3.6. Standaryzacja danych**

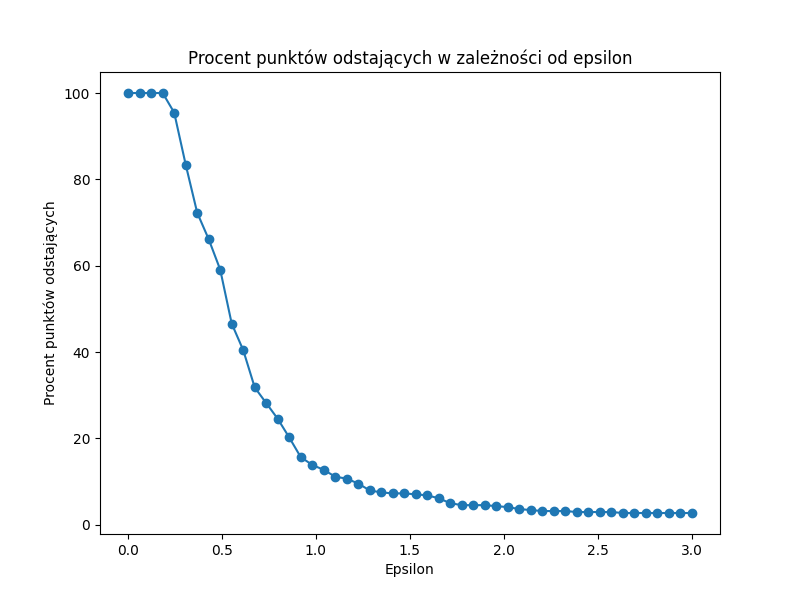
Dane zostały zestandaryzowane (z wyłączeniem kolumn 'Channel' i 'Region'), aby wszystkie cechy miały średnią zero i odchylenie standardowe jeden. Było to konieczne przed zastosowaniem algorytmu DBSCAN.

#### **3.3.7. Analiza punktów odstających przy użyciu DBSCAN**

* Przeprowadzono analizę liczby punktów odstających w zależności od wartości parametru epsilon (eps).
* Parametr eps zmieniał się od 0.001 do 3, przy 50 równomiernie rozłożonych wartościach.
* Parametr min\_samples ustawiono na dwukrotność liczby cech.
* Dla każdej wartości eps obliczono procent punktów zaklasyfikowanych jako wartości odstające.

#### **3.3.8. Wykres procenta punktów odstających w funkcji epsilon**

Utworzono wykres liniowy przedstawiający zależność procenta punktów odstających od wartości eps. Wykres pomógł w wyborze optymalnej wartości eps, która balansuje między liczbą klastrów a liczbą punktów odstających.



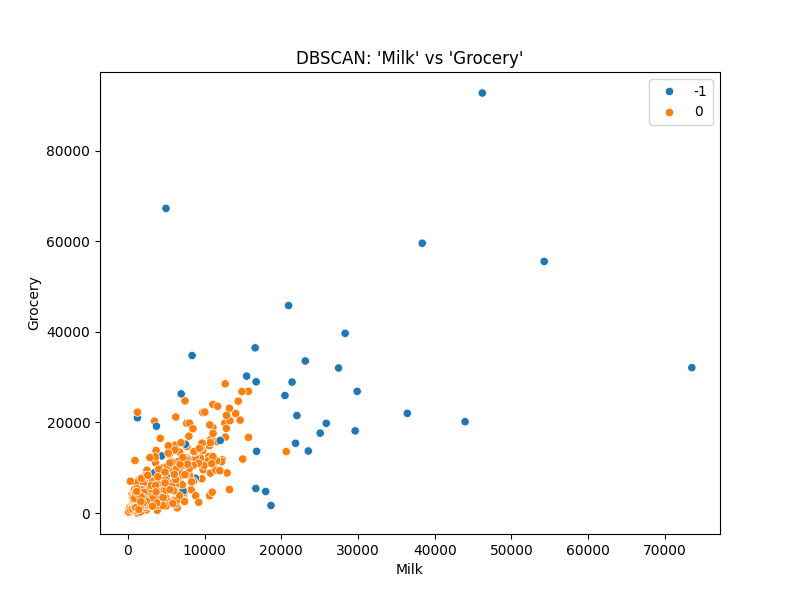
#### **3.3.9. Ponowna nauka modelu DBSCAN z odpowiednią wartością epsilon**

Na podstawie wykresu wybrano optymalną wartość eps wynoszącą 1,225 i ponownie przeprowadzono klasteryzację przy użyciu algorytmu DBSCAN.

#### **3.3.10. Wykresy punktowe z etykietami DBSCAN**

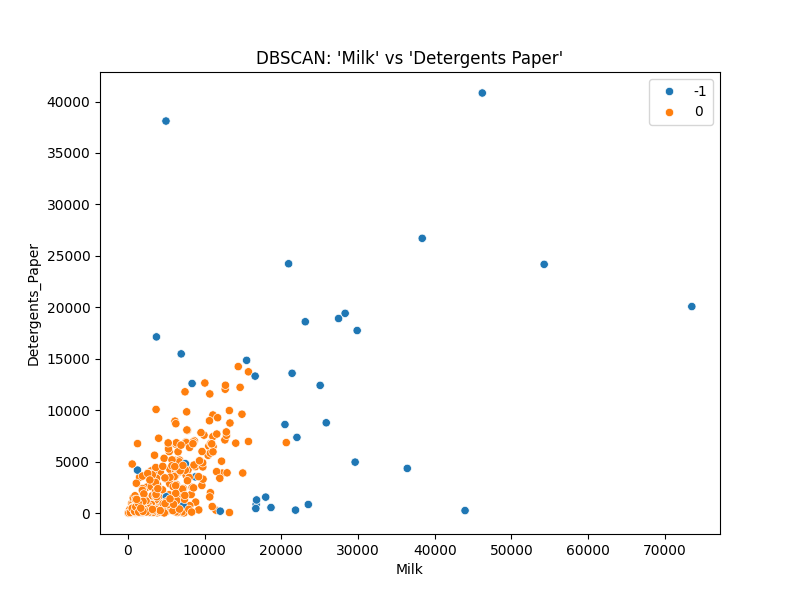
a. **'Milk' vs 'Grocery'**

Wykres przedstawia zależność między wydatkami na mleko a artykuły spożywcze, różnicując punkty na podstawie etykiet nadanych przez algorytm DBSCAN.



b. **'Milk' vs 'Detergents\_Paper'**

Podobnie jak wyżej, ale dla wydatków na mleko i artykuły papiernicze/detergenty.



#### **3.3.11. Analiza klastrów**

* Dodano kolumnę 'Labels' z etykietami klastrów nadanych przez DBSCAN.
* Usunięto kolumny 'Channel' i 'Region', aby skupić się na wydatkach.
* Porównano średnie wydatków w klastrach i punktach odstających.

#### **3.3.12. Identyfikacja kategorii z największymi różnicami**

Obliczono wariancję wydatków w poszczególnych kategoriach między klastrami. Kategoria z największą wariancją wskazuje na największe różnice w zachowaniach zakupowych między klastrami.

**Wynik:**

Kategoria wydatków z największymi różnicami między klastrami to **'Grocery'**.

**Dyskusja:**

Analiza wykazała, że wydatki na kategorię 'Grocery' różniły się najbardziej między klastrami, co sugeruje, że ta kategoria jest kluczowa w segmentacji klientów. Może to wskazywać na różne strategie zakupowe lub potrzeby klientów w zależności od klastra

# **Podsumowanie**

W projekcie zastosowano różne algorytmy klasteryzacji do analizy trzech zbiorów danych:

* **Segmentacja obrazu**: Algorytm k-średnich zredukował liczbę kolorów, segmentując obraz na podstawie podobieństwa kolorów. Pozwoliło to na uproszczenie obrazu i może być wykorzystane w kompresji czy analizie obrazów.
* **Klasteryzacja irysów**: Zarówno algorytmy hierarchiczne, jak i k-średnich z powodzeniem podzieliły dane na skupienia odpowiadające rzeczywistym gatunkom. Standaryzacja danych poprawiła jakość klasteryzacji.
* **Grupowanie klientów hurtowych**: Algorytm DBSCAN pozwolił na identyfikację klastrów oraz punktów odstających, ujawniając różnice w wydatkach na różne kategorie produktów. Analiza pokazała, które kategorie wydatków są najbardziej różnicujące dla segmentacji klientów.

# **Bibliografia**

* Pexels. (2020). Silhouette of Palm Trees Near Shoreline. Pobrano z https://www.pexels.com/photo/silhouette-of-palm-trees-near-shoreline-461940/
* Dua, D., & Graff, C. (2019). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. Dostępne na: <http://archive.ics.uci.edu/ml>
* Pedregosa, F., Varoquaux, G., Gramfort, A., i in. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
* Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90–95.
* McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 51–56.
* Scikit-learn: Machine Learning in Python. Dostępne na: <https://scikit-learn.org/>
* Seaborn: Statistical Data Visualization. Dostępne na: https://seaborn.pydata.org/
* OpenCV: Open Source Computer Vision Library. Dostępne na: <https://opencv.org/>